

# Porting a Web-Based Mapping Application to a Smartphone App\*

Hanan Samet  
hjs@cs.umd.edu

Marco D. Adelfio  
marco@cs.umd.edu

Brendan C. Fruin  
bfruin@umiacs.umd.edu

Michael D. Lieberman  
codepoet@cs.umd.edu

Benjamin E. Teitler  
bteitler@umiacs.umd.edu

Center for Automation Research, Institute for Advanced Studies,  
Department of Computer Science, University of Maryland  
College Park, MD 20742 USA

## ABSTRACT

NewsStand is a Web-based mapping application that we have developed to enable searching for spatially-referenced information by using a map query interface. Previously, we adapted the Web version to execute on mobile devices such as smartphones where the main issue that was confronted was how to deal with the considerably smaller display screen while retaining access to the application through the browser. In the current work we discuss the issues that we encountered in converting the Web-based application to a native App primarily on the iPhone and iPod Touch. These issues involve how to compensate for the absence of a hovering action as well as how to integrate an interaction restriction to one hand coupled with use of the thumb as the pointing mechanism. In addition, a significant effort is devoted to the implementation of an intuitive mechanism for undoing the most recent actions. Other issues include the formulation of navigation shortcuts to avoid excessive traffic with the supporting database.

## Categories and Subject Descriptors

H.3.1 [Information Storage and Retrieval]: Content Analysis and Indexing; H.3.3 [Information Storage and Retrieval]: Information Search and Retrieval; H.5.2 [Information Interfaces and Presentation]: User Interfaces

## General Terms

Algorithms, Design, Human Factors, Performance

## Keywords

NewsStand, Touch Screen Gesturing Interface, Map Query Interface, Smartphone, App

## 1. INTRODUCTION

The NewsStand system [7, 10] (see also the related STEWARD [5] and TwitterStand [8] systems) is an example application of a general framework being developed at the

\*This work was supported in part by the National Science Foundation under Grants IIS-10-18475, IIS-09-48548, IIS-08-12377, CCF-08-30618, and IIS-07-13501.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

ACM SIGSPATIAL GIS '11, November 1–4, 2011, Chicago, IL, USA  
Copyright 2011 ACM 978-1-4503-1031-4/11/11 ...\$10.00.

University of Maryland at College Park (see also the related systems QUILT [9] and the SAND Browser [6]) to enable searching for information using a map query interface. When the information domain is news, the underlying search domain results from monitoring the output of over 8,000 RSS news sources and making it available for map-based retrieval within minutes of publication. The advantage of doing so is that a map, coupled with the ability to vary the zoom level at which it is viewed, provides an inherent granularity to the search process that facilitates an approximate search. This distinguishes it and our related efforts from today's conventional keyword-based search methods that provide a very limited facility for approximate searches realized primarily by permitting a match via use of a subset of the keywords. However, it is often the case that users do not have a firm grasp of which keyword to use, and thus would welcome the capability for the search also to take synonyms into account. For queries to spatially-referenced data, the map query interface is a step in this direction as the act of pointing at a location (e.g., by the appropriate positioning of a pointing device) and making the interpretation of the precision of this positioning specification dependent on the zoom level is equivalent to permitting the use of spatial synonyms.

NewsStand was originally designed as a Web mapping application to be run on desktops and laptops where the interactions use a mouse pointing device. In this paper, we describe some of the issues that we faced when porting it to run on smartphones (e.g., the iPhone and iPod Touch as well as Android) where it runs as a native App (rather than as a Web application [2, 7] and where the interactions are solely via touch gestures using a one-handed interface with the thumb as the pointing mechanism. We also discuss a mechanism for undoing the most recent actions.

## 2. ISSUES

### 2.1 General Query Support

NewsStand supports two types of queries: location-based and feature-based [1] variants of the top- $k$  query where “top” is a measure of importance with respect to news topics which combines number of occurrences with velocity and acceleration of the rate at which they occur. The location-based variant returns the top  $k$  topics that mention a specific spatial region, which is usually the part of the world visible on the display screen. The actual output is in the form of a map containing markers at the principal locations where the top  $k$  topics occur (see Figure 1(a) and termed “Map Mode”). Tapping on a marker reveals a headline associated with the corresponding location (“Libya” in Figure 1a); tapping on the headline reveals the remaining headlines (Figure 2(a)); while tapping on each headline reveals more detail about the headline including an image, a snippet of text from the corresponding article that mentions the location, a description

of the article, as well as icons denoting the presence of links to associated images, videos, and other articles on the same topic (Figure 2(b)).

The feature-based variant simply returns the top  $k$  locations mentioned in articles about topic  $T$  or just article  $Y$ , which are detected by techniques such as in [4]. The actual output has two components. The first consists of brief details on the topic or article including a link to a representative article of the topic or the article itself, as well as icons denoting the presence of links to associated images, videos, and other articles on the same topic. The second is a map containing markers at the locations mentioned in the topic/article and hence are likely to occur (see Figure 1(b) and termed “Top Stories Mode”). Several topics or articles can be viewed depending on where on the output component the user is hovering (see Section 2.2). The value of  $k$  is controlled by moving a slider. The topics are ordered either by importance, or by time in terms of the one with the most recent article. A refresh icon button on the map (regardless of the mode) updates the map with the latest streaming news which enables users to remain current.



Figure 1: iPhone and iPod Touch screen shots showing (a) Map Mode once the marker at “Libya” has been tapped, and (b) Top Stories Mode assuming a left-handed one-handed preference setting. In the case of Map Mode, the markers are restricted to topics with a keyword of “Obama” and in articles in “Reuters”.

The large display screen available on the Web version uses menus and several buttons for control purposes. On the other hand, the App version has a very small display screen that limits the use of buttons and menus, which means that it is important to choose the right ones. Regardless of the mode of operation, the bottom of the display screen contains a row of buttons including controls for toggling between “Map Mode” and “Top Stories Mode”, as well as one for toggling between NewsStand and its companion TwitterStand [8] system which uses the same user interface. Other buttons include “Info” providing a link to NewsStand’s online documentation; “Sources” enabling restricting the news source from which the representative article displayed for a topic is drawn (this includes a collection of user specified sources, a subset of predefined sources known to be reputable, and the most recent article about the topic regardless of source); and “Search” enabling searching for a specific topic by providing the appropriate keyword. The “Settings” button enables personalizing use of the system to

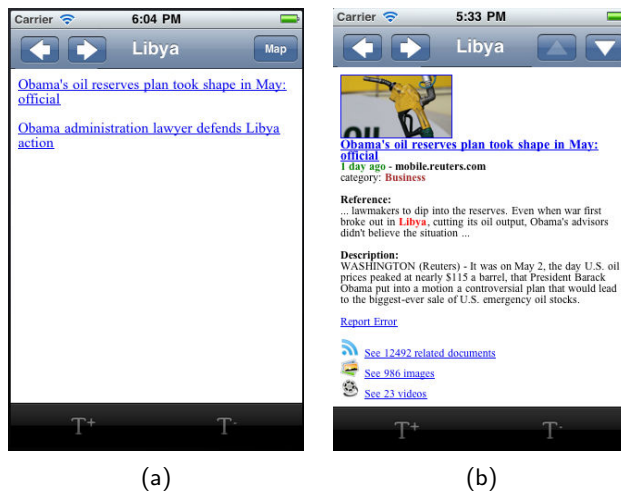


Figure 2: (a) Headlines corresponding to Libya in Figure 1(a) and (b) a snippet corresponding to the first headline in (a).

include a “Home” location such as where the user hails from so that news from this location can be obtained, restricting the type of the topics such as General News, Business, Science and Technology, Entertainment, Health, and Sports, as well as setting filters to limit the displayed topics based on the number of images and/or videos associated with them.

The top row enables a more efficient use of “Map Mode” so users can proceed directly to locations of interest, bypassing navigation in the map interface using zooming and the redrawing that this may entail. This includes a user-settable “Home”, “Local” denoting the user’s current location, “World” indicating the entire world, and “Locate” where a user inputs a specific location. A slider enables varying of the number of locations for which topics are displayed.

The navigation buttons result in one-time actions. On the other hand, the actions of the remaining buttons are in effect until they are explicitly reset. In the case of the “Sources” and “Search” buttons, this has caused great confusion among users who forget that they are still set. Thus in order to remedy this problem, the settings of these buttons are displayed explicitly on the screen with an “X” to their right which is tapped in order to disable them (e.g., a search keyword of “Obama” and the source “Reuters” in Figure 1(a)). In the case of multiple sources, users can see them by tapping their names which results in cycling through them. Tapping the “X” of the name of a visible source disables it.

## 2.2 Hovering

The Web version was designed primarily for use with a mouse pointing device, which allows two types of interaction:

1. detecting that the mouse hovers over a location on the display screen,
2. detecting a click on the same location.

For the versions that make use of a gesturing touch interface, the effect of a click action is achieved by a tapping action, while the hovering mode of interaction is not available in its full generality as there is no device whose motion is to be tracked. Instead, the effect of hovering is achieved by a combination of an explicit scrolling or panning action (the difference depends on whether the motion is vertical or horizontal, respectively) coupled with an implicit tapping action, explained below. In particular, we observe that the hovering action is similar to a combination of scrolling and panning actions with the difference that when using a pointing device such as a mouse, it is the pointing device that moves,

while for a gesturing touch interface, it is the underlying data elements on the display screen that move. The result is what we term a “wheel of fortune” graphical user interface, analogous to the casino game of the same name, where the wheel corresponds to the data on the display screen, the segments of the wheel correspond to the topics, the pointer is a fixed location on the screen over which the topics move, and the winner is the topic that overlaps the pointer when the scrolling stops.

Below we describe the implementation of hovering on the smartphone web version. We make use of the Mobile Safari Web browser<sup>1</sup> which mimics the effect of the hovering action (which on a conventional desktop and laptop occurs once you move the pointing device) by registering the location on the display screen where the hovering action is to take place. This is done by firing the equivalent of a Javascript “mouseover” event at the location on the display screen where a tap on a data element occurred and which registers the location. This is followed by repeated firings of the “mouseover” equivalent event (i.e., resulting in what we term an “implicit tapping action”) as data elements are moved (via, without loss of generality, the gesturing scrolling action) under the location of the last (i.e., immediately preceding) tap. This is done until the motion ceases at which time the final equivalent of a Javascript “mouseover” event is fired, which also fires an event corresponding to a tap even though no explicit tap took place. The appropriate action is taken for this implicit tap which depends on the context in which the original tap and scroll gestures took place. Notice that this is equivalent to moving the pointing device either manually or by scrolling using a mouse wheel or the down and up arrow keys. The app version is even simpler as we make use of the built-in table structure of iOS in order to store the relative position of the user’s last selection when scrolling stopped. Now, when the the table detects a subsequent scroll gesture, the stories are updated and the table cell in the stored position is implicitly tapped when the scroll gesture terminates.

### 2.3 One-Handed Interaction

The user interface of the App version of NewsStand for the smartphone is designed so that all interactions can be achieved using just one hand (e.g., [3]) when used in portrait mode (although it can also be used in landscape mode, in which case both hands are used), thereby enabling its use both when walking and standing. This frees the other hand to carry items or interact with the environment (e.g., most importantly, writing with one hand while manipulating the smartphone with the other hand). Observe that performing a direct port of the Web version from the desktop or laptop to the smartphone where the screen display is much smaller means that the interaction must be very precise, which can only be accomplished by use of a stylus-like device. However, this method of interaction requires two hands (one to hold the smartphone and one to hold the stylus), as well as most likely the use of a magnifying glass, or at least some very powerful glasses, to retain the full functionality of the Web application. On the other hand, one-handed interaction results in use of the thumb as a pointing device, which has the drawback of low precision, thereby leading to large error rates when trying to access objects that are relatively small due to finger occlusion as well as being out of reach. Simply making the objects larger wastes display screen space and possibly precision due to their being too proximate. Thus considerable attention must be paid to minimizing needless features (often implemented by buttons on the top and bottom rows of the display screen as discussed in Section 2).

NewsStand has been ported so that when in portrait mode,

the smartphone can be held with one hand resting on the hand’s palm and supported by the four non-thumb fingers, while all actions can be performed with the thumb which serves as the pointing device. Areas that are out of easy reach are accessed by using the index finger to slide the smartphone up and down along the hand’s palm while releasing the three remaining non-thumb fingers. Note that the one-handed interaction stipulation precludes the use of a pinching action for zooming-in and zooming-out on the maps which require two fingers to move towards each other and away from each other, respectively (although, of course, it still can be used; it just can’t be done with one finger). Therefore, we have added “+” and “-” buttons on the maps (see Figure 1) which achieve these two actions, respectively, by simply tapping them with the thumb. The ease of use in terms of reach of these buttons depends, in part, on the hand in which the smartphone is held and thus is made user-settable by use of the “One-Handed Zoom Preference” feature as part of the options available in the “Settings” button where “Left” (“Right”) handed use positions the “+” and “-” buttons on the right (left) side one above the other for “Top Stories Mode” and at an angle for “Map Mode” (see Figure 1 which assumes a “One-Handed Zoom Preference” setting of “Left”) so that the thumb must travel the same distance to each of the buttons, thereby lessening the likelihood that the wrong one will be tapped. Note that the intent is for the preference to be chosen on the basis of increasing the utility of the free hand, rather than the hand that holds the smartphone, which really has nothing to do other than move the thumb! Thus right (left)-handed people should use the “Left” (“Right”) preference setting as, for example, this will enable them to write with the free hand. Similar considerations lead to placement of the “Done” button in “Settings” and “Sources” in the upper left (right) corner of the display screen for left (right)-handed users.

The above solution is fine when the underlying area is a map and does not have too many other tappable points (such as the markers at the locations mentioned in articles in “Top Stories Mode” and the locations where the top topics in “Map Mode” occur in which case the underlying map can be panned or scrolled with the thumb to avoid the overlap before activating the zoom action). However, the situation becomes more complex when the underlying display screen contains many links as is the case when viewing text such as news articles (i.e., their HTML pages), the set of headlines associated with a location, as well as the article snippets associated with a headline. In this case, the default zoom-in action is not what we want as in order to read the page we will have to pan it constantly. Instead, we have adopted a solution used in many Apps which incorporates a pair of buttons in the bottom row of the display screen labeled “T+” and “T-” that enable users to zoom in and out, respectively, on the actual text, thereby decoupling the zoom from the links. This also reformats the webpage so that lines don’t wrap around, which avoids the need to pan.

### 2.4 Undoing Actions

When performing actions in a user interface, it is often desirable to undo the most recent action. For example, clicking a story marker in NewsStand’s interface will display more information about the story. When finished viewing the story, the user may want to undo the click action and return to the map. This act of undoing the most recent action is often implemented by means of a *back button*. In effect, the back button acts as a navigator between application states. Performing an action creates a new application state; clicking the back button returns to the previous application state. Therefore, a functional back button requires a way to store, and restore, the application’s state from a history of states.

Both NewsStand’s Web and App versions feature back

<sup>1</sup><http://www.apple.com/iphone/features/safari.html>

buttons. However, their implementations, and how they manage the application's state, differ considerably. The Web version leverages the back button built into Web browsers, which navigates backward through the browser's webpage history. In effect, the webpage history serves as the browser's state information. Normally, clicking a link on a webpage loads a new page, which is then added to the webpage history. Clicking the back button at that point would navigate backward to the previous page. More precisely, when the browser's URL (in the location bar) changes, the new URL is added to the browser's webpage history. However, because NewsStand's interactive functionality is implemented via AJAX, which modifies the rendered webpage but does not change its URL, actions taken in NewsStand would not normally be added to the browser's webpage history. As a result, the back button does not always function like an "undo" without special handling.

NewsStand's Web version leverages the browser's back button by using a workaround typical of AJAX-based Web applications, namely encoding NewsStand's state into the URL by setting the URL's *location hash*. The location hash is the portion of a page's URL that follows a hash symbol ("#"), and was originally intended to allow links to named portions of a page by placing the name after the hash symbol. If the named portion does not exist on the webpage, then nothing apparently happens. However, because the URL changes, the new URL including the location hash is saved into the browser's history, which is navigable via the back button. Due to these effects, the location hash can be used for navigation within AJAX-based applications by storing the application state as a location hash string. This state saving is accomplished by a process called *serialization*, which converts a set of state variables and their values (e.g., "x:1", "y:3", "z:5") to a string representation (e.g., "x=1&y=3&z=5"). This serialized representation of the application state can be stored in the location hash and used in navigation. When the back button is used and a previous location hash is loaded, a reverse process of *deserialization* can restore the application's state variables by parsing the state string. While this method can function, it is awkward to use because it uses the location hash in a way for which it was not intended, and requires the manual implementation of state variables, and serialization and deserialization routines for these variables. These implementations can be quite cumbersome since the application's state space increases very rapidly with each additional interface feature.

Mobile Apps also usually contain back buttons, but their implementation is rather different. Instead of modifying the application state with each action, application states are segmented and layered into what is termed a *view stack*. Such a stack consists of a hierarchy of *views*, which are essentially application windows that accept input actions. Only the topmost view in the hierarchy is activated and visible to accept inputs. Performing an action in the topmost view will generally create a new view and push it onto the stack. Clicking the back button pops the topmost view from the stack and makes the lower view visible. For example, NewsStand's default or base view in its mobile App is its map interface, i.e., its map view, which is at the lowest level of the hierarchy. When entering an action into the map view (e.g., clicking a story marker on the map), a new view is created containing the marker information, and pushed onto the view stack, making the marker information visible. Clicking the back button pops the marker view from the stack, removes it, and reactivates the map view underneath.

It is important to observe that views in the mobile App version are modular and independent, so adding a new view to the stack does not affect the views underneath, allowing them to be restored quickly and easily when the topmost view is popped from the stack. This stands in contrast to

the Web version, where the entire application's state changes with each action, and thus must be restored as described above. However, the view functionality does have an analog in Web browsers, namely *tabs*. When opening a new browser tab, the effect is to create a new "view" that can be modified through actions such as visiting webpages and clicking links. Any actions in the new tab do not affect the other tabs, in much the same way as actions in the topmost view do not affect other views in the stack. In the Web browser, the analogous action to popping a view from the stack would be closing the tab, which restores the previously-open tab. Unfortunately, browser tabbing cannot be harnessed for use with the browser back button, which is not programmable and only operates on browser history, so it will not close tabs. While a hindrance, this may be needed to avoid scam websites that could maliciously reprogram the back button. Nonetheless, it greatly complicates "undo" actions in the Web version, as compared to the App version. In other words, when we have a tab, the back button cannot be used to implicitly return to the state which opened the tab, thereby closing the tab. In addition, although the tabs are invoked in a stack-like manner in the Web version, the tabs are active at all times, while this is not the case for the entries in the stack of the App version (i.e., no concurrency).

### 3. CONCLUSION

We presented an overview of some of the issues that we faced in porting a Web-based map application to a smartphone App with a gesturing touch screen interface such as the iPhone and iPod Touch. The main difference from our prior work [7] is that an entirely new user interface was developed based on a stipulation that all interactions could be performed with one hand with the hand's thumb serving as the pointing device. In addition, the App freed us from some of the interaction restrictions for undoing actions that arose from the manner in which Web applications are built. Due to procedural issues in getting Apps approved, the App is not yet in the App store. However, the Web version can be found at <http://newsstand.umiacs.umd.edu>.

### 4. REFERENCES

- [1] W. G. Aref and H. Samet. Efficient processing of window queries in the pyramid data structure. In *PODS'90*, pages 265–272, Nashville, TN, Apr. 1990.
- [2] A. Charland and B. LeRoux. Mobile application development: web vs. native. *CACM*, 54(5):49–53, May, 2011.
- [3] A. K. Karlson and B. B. Bederson. One-handed touchscreen input for legacy applications. In *CHI'08*, pages 1399–1408, Florence, Italy, Apr. 2008.
- [4] M. D. Lieberman and H. Samet. Multifaceted toponym recognition for streaming news. In *SIGIR'11*, pages 843–852, Beijing, China, July 2011.
- [5] M. D. Lieberman, H. Samet, J. Sankaranarayanan, and J. Sperling. STEWARD: architecture of a spatio-textual search engine. In *GIS'07*, pages 186–193, Seattle, WA, Nov. 2007.
- [6] H. Samet, H. Alborzi, F. Brabec, C. Esperança, G. R. Hjaltason, F. Morgan, and E. Tanin. Use of the SAND spatial browser for digital government applications. *CACM*, 46(1):63–66, Jan. 2003.
- [7] H. Samet, B. E. Teitler, M. D. Adelfio, and M. D. Lieberman. Adapting a map query interface for a gesturing touch screen interface. In *WWW'11 (Companion Volume)*, pages 257–260, Hyderabad, India, Mar.-Apr. 2011.
- [8] J. Sankaranarayanan, H. Samet, B. Teitler, M. D. Lieberman, and J. Sperling. TwitterStand: News in tweets. In *GIS'09*, pages 42–51, Seattle, WA, Nov. 2009.
- [9] C. A. Shaffer, H. Samet, and R. C. Nelson. QUILT: a geographic information system based on quadtrees. *IJGIS*, 4(2):103–131, Apr.–Jun. 1990.
- [10] B. Teitler, M. D. Lieberman, D. Panozzo, J. Sankaranarayanan, H. Samet, and J. Sperling. NewsStand: A new view on news. In *GIS'08*, pages 144–153, Irvine, CA, Nov. 2008.